

IN THE CLAIMS:

1. (Currently Amended) A method of routing data packets to a queue pair, comprising:

receiving a data packet having a header in which one or more [[IP]] Internet Protocol filter values are identified;

identifying a queue pair in a plurality of queue pairs based on the one or more Internet Protocol filter values in the header of the data packet, wherein a single channel adapter supports the plurality of queue pairs; and

routing the data packet to the identified queue pair.

2. (Currently Amended) The method of claim 1, wherein identifying the queue pair includes:

generating a hash value based on the one or more Internet Protocol filter values; and

retrieving a hash table entry based on the hash value.

3. (Original) The method of claim 2, further comprising:

determining if a collision bit in the hash table entry is set; and

retrieving a collision table entry corresponding to the hash table entry if the collision bit is set.

4. (Currently Amended) The method of claim 3, further comprising:

comparing the one or more Internet Protocol filter values in the header of the data packet header to filter values in the collision table entry; and

identifying the queue pair based on the comparison of the one or more Internet Protocol filter values in the header of the data packet header to the filter values in the collision table entry.

5. (Original) The method of claim 1, wherein the method is implemented in a host channel adapter set up to support filtering.

6. (Original) The method of claim 5, wherein the host channel adapter is set up to support filtering by using a Modify HCA verb to enable filtering in the host channel adapter.
7. (Original) The method of claim 1, wherein the queue pair is a queue pair that is set up to support filtering by using a Modify QP verb to enable filtering.
8. (Original) The method of claim 7, wherein the Modify QP verb identifies the filter value for each filter type enabled from filter types supported by a corresponding host channel adapter.
9. (Currently Amended) The method of claim 1, wherein the one or more Internet Protocol filter values are Internet Protocol over InfiniBand transport and/or network layer filter values.
10. (Currently Amended) The method of claim 1, wherein identifying [[a]] the queue pair in a plurality of queue pairs based on the one or more Internet Protocol filter values in the header of the data packet includes using a content addressable memory.
11. (Currently Amended) A computer program product in a computer readable medium for routing data packets to a queue pair, comprising:
first instructions for receiving a data packet having a header in which one or more [[IP]] Internet Protocol filter values are identified;
second instructions for identifying a queue pair in a plurality of queue pairs based on the one or more Internet Protocol filter values in the header of the data packet, wher cin a single channel adapter supports the plurality of queue pairs; and
third instructions for routing the data packet to the identified queue pair.
12. (Currently Amended) The computer program product of claim 11, wherein the second instructions for identifying the queue pair include:

instructions for generating a hash value based on the one or more Internet Protocol filter values; and

instructions for retrieving a hash table entry based on the hash value.

13. (Original) The computer program product of claim 12, further comprising:
instructions for determining if a collision bit in the hash table entry is set; and
instructions for retrieving a collision table entry corresponding to the hash table entry if the collision bit is set.

14. (Currently Amended) The computer program product of claim 13, further comprising:

instructions for comparing the one or more Internet Protocol filter values in the header of the data packet header to filter values in the collision table entry; and

instructions for identifying the queue pair based on the comparison of the one or more Internet Protocol filter values in the header of the data packet header to the filter values in the collision table entry.

15. (Original) The computer program product of claim 11, wherein the computer program product is executed in a host channel adapter set up to support filtering.

16. (Original) The computer program product of claim 15, wherein the host channel adapter is set up to support filtering by using a Modify HCA verb to enable filtering in the host channel adapter.

17. (Original) The computer program product of claim 11, wherein the queue pair is a queue pair that is set up to support filtering by using a Modify QP verb to enable filtering.

18. (Original) The computer program product of claim 17, wherein the Modify QP verb identifies the filter value for each filter type enabled from filter types supported by a corresponding host channel adapter.

19. (Currently Amended) The computer program product of claim 11, wherein the one or more Internet Protocol filter values are Internet Protocol over InfiniBand transport and/or network filter values.
20. (Currently Amended) The computer program product of claim 11, wherein the second instructions for identifying [[a]] the queue pair in a plurality of queue pairs based on the one or more Internet Protocol filter values in the header of the data packet include instructions for using a content addressable memory.
21. (Currently Amended) An apparatus for routing data packets to a queue pair, comprising:
 - means for receiving a data packet having a header in which one or more [[IP]] Internet Protocol filter values are identified;
 - means for identifying a queue pair in a plurality of queue pairs based on the one or more Internet Protocol filter values in the header of the data packet, wherein a single channel adapter supports the plurality of queue pairs; and
 - means for routing the data packet to the identified queue pair.
22. (Currently Amended) The apparatus of claim 21, wherein the means for identifying the queue pair includes:
 - means for generating a hash value based on the one or more Internet Protocol filter values; and
 - means for retrieving a hash table entry based on the hash value.
23. (Original) The apparatus of claim 22, further comprising:
 - means for determining if a collision bit in the hash table entry is set; and
 - means for retrieving a collision table entry corresponding to the hash table entry if the collision bit is set.

24. (Currently Amended) The apparatus of claim 23, further comprising:
means for comparing the one or more Internet Protocol filter values in the header of the data packet header to filter values in the collision table entry; and
means for identifying the queue pair based on the comparison of the one or more Internet Protocol filter values in the header of the data packet header to the filter values in the collision table entry.

25. (Original) The apparatus of claim 21, wherein the apparatus is part of a host channel adapter set up to support filtering.

26. (Original) The apparatus of claim 25, wherein the host channel adapter is set up to support filtering by using a Modify HCA verb to enable filtering in the host channel adapter.

27. (Original) The apparatus of claim 21, wherein the queue pair is a queue pair that is set up to support filtering by using a Modify QP verb to enable filtering.

28. (Original) The apparatus of claim 27, wherein the Modify QP verb identifies the filter value for each filter type enabled from filter types supported by a corresponding host channel adapter.

29. (Currently Amended) The apparatus of claim 21, wherein the one or more Internet Protocol filter values are Internet Protocol over InfiniBand transport and/or network filter values.

30. (Currently Amended) The apparatus of claim 21, whercin the means for identifying [[a]] the queue pair in a plurality of queue pairs based on the one or more Internet Protocol filter values in the header of the data packet includes means for using a content addressable memory.